МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет
«Харківський політехнічний інститут»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторної роботи
«Використання одновимірних масивів у програмах мовою Delphi»

з курсу «Програмування»
для студентів напряму 6.040302 – Інформатика
(спеціалізація «Соціальна інформатика»)

Затверджено редакційно-видавничою
радою університету,
протокол № 3 від 28.12.09.

Харків НТУ «ХПІ» 2010

Методичні вказівки до лабораторної роботи «Використання одновимірних масивів у програмах мовою Delphi» з курсу «Програмування» для студентів напряму 6.040302 – Інформатика (спеціалізація «Соціальна інформатика») / Уклад. М. І. Безменов. – Х. : НТУ «ХПІ», 2010. – 17 с.

Укладач М. І. Безменов

Рецензент Л. М. Любчик

Кафедра системного аналізу і управління

Мета роботи

Освоєння сфери застосування одновимірних масивів та методів розв'язання задач з програмування з їх використанням.

1. ТЕОРЕТИЧНІ ОСНОВИ

У Delphi існує механізм, що дуже просто розв'язує проблему позначення великої кількості однорідних елементів. При цьому можна не обмежуватися однією змінною, у яку по черзі заноситься множина значень, або оголошувати велику кількість різнойменних змінних. Альтернативою є використання ***масиву*** – змінної типу **`array`**, яка містить визначене число однойменних елементів одного й того самого типу і дозволяє посилатися на перший, другий і всі наступні елементи за значенням їх індексів.

Для опису так званого одновимірного масиву потрібно, крім його імені, зазначити межі зміни значення індексу та тип елементів, що є спільним для всіх елементів масиву:

```
var
  ім'я: array [нижня_межа_інд..верхня_межа_інд] of тип_елементів;
```

Тут ім'я – ім'я масиву;

нижня_межа_інд – найменше можливе значення індексу;

верхня_межа_інд – найбільше можливе значення індексу;

тип_елементів – спільний тип елементів масиву.

Звертаємо увагу на наступне:

- нижня та верхня межі зміни індексу задаються тільки константами або виразами над константами;
- тип нижньої та верхньої меж може бути будь-яким дискретним типом, тобто таким типом, у якому, по-перше, визначено дискретну послідовність значень і, по-друге, кожне з цих значень можна перерахувати по порядку (це будь-який цілочисловий тип, булев тип, символьний тип, перелічений тип);
- використання змінних при задавання меж заборонене;
- повинна виконуватися нерівність нижня_межа_інд ≤ верхня_межа_інд;
- найчастіш за все діапазон зміни індексів задається цілочисловими значеннями;
- елементи масива мають один і той самий тип (тип_елементів), причому цей тип може бути довільним, за винятком деяких обмежень.

Приклади опису масивів:

```
var
  a, b: array[-10..100] of Integer;
  c   : array['a'..'z'] of Boolean;
```

Діапазон зміни індексу не може перевищувати 2 Гбайт (у випадку багатовимірних масивів це стосується кожного з вимірів). Таке ж саме обмеження має місце і для загального обсягу пам'яті, займаної масивом.

Для звертання до елемента масиву вказують ім'я масиву й у квадратних дужках індекс, значення якого визначає розташування елемента усередині масиву. Індексувати можна як константами і змінними, так і виразами, результат обчислення яких дає значення дискретного типу, що є сумісним за присвоюванням з типом меж.

Коли індекс масиву може набувати всіх можливих значень деякого дискретного типу, тоді при описі масиву доцільно задавати ім'я типу замість меж зміни індексу (при цьому межами індексу будуть перше та останнє значення в описі типу індексу).

Межі зміни індексів можуть задаватися з використанням раніше оголошених констант.

Рекомендується попередньо оголошувати масивний тип у розділі опису типів, а потім вживати його для опису масивів:

```
const
  Num = 150;
type
  DaysOfWeek = (Monday, Tuesday, Wednesday,
                Thursday, Friday, Saturday,
                Sunday);          //Перелічений тип
  Years = 1991..2010;                    //Відрізок
  TTax = array[Years] of Real;          //Тип-масив
var                    //Можливі методи опису масиву
  Gain: array[DaysOfWeek] of Real;
  TaxOfYear: TTax;
  Production: array[1..Num] of string[10];
```

У наведеному прикладі тип `TTax` визначає масив, індекси якого можуть набувати значень від `1991` до `2010`. За допомогою цього типу оголошено масив `TaxOfYear`. Для індексування елементів масиву `Gain` має використовуватися значення переліченого типу `DaysOfWeek`, причому сам масив має 7 елементів відповідно до опису типу `DaysOfWeek`.

Функція SizeOf, застосована до імені масиву або імені масивного типу, повертає кількість байт, що відводиться під масив. Так для наведених вище описів, якщо змінна Size має тип Integer, то після виконання, як оператора

```
Size := SizeOf(TaxOfYear),
```

так і оператора

```
Size := SizeOf(TTax),
```

вона матиме значення 88.

Функції Low, High і Length, застосовані до імені масиву, повертають відповідно мінімальне значення індексу, максимальне значення індексу й кількість елементів масиву. Для описаного вище масиву TaxOfYear згадані функції дадуть такі результати:

```
Low(TaxOfYear) = 1991,
High(TaxOfYear) = 2010,
Length(TaxOfYear) = 20.
```

При описі масиву і звертанні до його елементів квадратні дужки можуть бути замінені парою складених символів (. та .).

2. ПРИКЛАДИ ПРОГРАМ

Приклад 1. Які різні цифри входять у ціле число *N*?

Розмістимо на формі однорядковий редактор TEdit (надамо йому ім'я edInput1) для введення числа *N*, багаторядковий редактор TMemo (надамо йому ім'я mmOutput1) для виведення результату та кнопку Button1. В такому разі задачу розв'язує такий оброблювач події OnClick кнопки Button1:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  N: Int64;                              // Не більше 18 цифр!
  Numerals: array [1..19] of Byte;              // Різні цифри
  i,                                     // Параметр циклу for
  QuantityOfNumerals,                 // Кількість різних цифр
  Numeral:Byte;                            // Виділювана цифра
  Flag: Boolean;
begin
  N := StrToInt64(edInput1.Text);
  QuantityOfNumerals := 0;     // Поки що немає жодної цифри
  repeat
    Numeral := Abs(N mod 10);         // Одержуємо нову цифру
    Flag := False;// Вважаємо, що цифра раніше не виділялася
```

```
            // Перевіряємо, чи виділялася раніше така цифра
  for i := 1 to QuantityOfNumerals do
    if Numeral = Numerals[i] then begin
      Flag := True;           // Раніше така цифра вже була!
      Break;                                  // Вихід з циклу
    end;
    if Flag = False then begin     // Краще так: if not Flag
      QuantityOfNumerals := QuantityOfNumerals + 1;
      Numerals[QuantityOfNumerals] := Numeral;
    end;
    N := N div 10;// Відтинаємо останню цифру в запису числа
  until N = 0;
  mmOutput1.Lines.Add('Число ' + edInput1.Text +
                      ' містить такі різні цифри:');
  for i := 1 to QuantityOfNumerals do
    mmOutput1.Lines.Add(IntToStr(Numerals[i]));
end;
```

Зауважимо, що в програмі цифри виділяються, починаючи від самої правої з них у запису числа. Аналогічно вони і виводяться. Щоб змінити порядок виведення на протилежний, останній цикл **for** потрібно записати в такий спосіб:

```
    for i := QuantityOfNumerals downto 1 do
      mmOutput1.Lines.Add(IntToStr(Numerals[i]));
```

Значно простішим є такий оброблювач події OnClick кнопки Button1:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  N: Int64;                               // Не більше 18 цифр!
  Amount: array [0..9] of Byte;   // Кількість входжень цифр
  i: Integer;                             // Параметр циклу for
begin
  N := StrToInt64(edInput1.Text);
  for i := 0 to 9 do           // Поки що вважаємо, що цифри
    Amount[i] := 0;                       // не входять в число
  repeat
    Inc(Amount[Abs(N mod 10)]);
    N := N div 10;// Відтинаємо останню цифру в запису числа
  until N = 0;
  mmOutput1.Lines.Add('Число ' + edInput1.Text+
                  ' містить такі різні цифри:');
  for i := 0 to 9 do
    if Amount[i] <> 0 then
      mmOutput1.Lines.Add(IntToStr(i));
end;
```

Приклад 2. Упорядкувати числовий масив, що вміщує не більш 20 дійсних чисел, за неубуванням методом «бульбашки».

Розмістимо на формі однорядковий редактор TEdit (надамо йому ім'я edInput1) для введення кількості елементів масиву *N*, багаторядковий редактор TMemo (надамо йому ім'я mmOutput1) для виведення результату та дві кнопки Button1 та Button2, перша з яких буде використовуватися для подання команди на введення даних, а друга – для подання команди на сортування масиву і виведення його остаточного вмісту. Кнопки розмістимо одну на одній, причому у властивість Visible кнопки Button2 запишемо занчення False. Запишемо також у властивості Caption кнопок Button1 та Button2 тексти Ввести і Сортувати відповідно. Над редактором edInput1 розмістимо мітку, надавши їй ім'я lbOutput1 та записавши у її властивість Caption текст Уведіть кількість елементів масиву.

Включимо також в опис класу TForm1 у розділ **public** опис двох полів:

```
N: Integer;                  //Кількість елементів масиву
a: array [0..19] of Real;                          //Масив
```

Відзначимо, що ці змінні можна описати у розділі опису змінних секції **implementation** модуля.

Для кнопок Button1 і Button2 створимо такі опрацьовувачі події OnClick:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  i: Integer;
begin
  DecimalSeparator := '.';
  N := StrToInt(edInput1.Text);        // Кількість елементів
  lbOutput1.Visible := False;        // Мітка тепер прихована
  for i := 0 to N - 1 do
    a[i] := StrToInt(InputBox('Введення масиву',
        'Уведіть елемент a[' + IntToStr(i + 1) + ']', '0'));
  mmOutPut1.Lines.Add('Початковий масив');   // Повідомлення
  for i := 0 to N - 1 do            //Виводимо початкові дані
    mmOutput1.Lines.Add('a[' + IntToStr(i + 1) + '] = ' +
                        FloatToStr(a[i]));
  edInput1.Visible := False;       // Однорядковий редактор і
  Button1.Visible := False; // кнопка Button1 тепер невидимі
  Button2.Visible := True; // Кнопка Button2 тепер є видимою
end;

procedure TForm1.Button2Click(Sender: TObject);
var
```

```
  i, j: Integer;
  b: Real;
begin                                          // Метод "бульбашки"
  for i := 0 to N - 2 do
    for j := N - 1 downto i + 1 do
      if a[j] < a[j - 1] then begin
        b := a[j];
        a[j] := a[j - 1];
        a[j - 1] := b;
      end;
  mmOutPut1.Lines.Add('Результуючий масив');
  for i := 0 to N - 1 do       //Виводимо результуючий масив
    mmOutput1.Lines.Add('a[' + IntToStr(i + 1) + '] = ' +
                        FloatToStr(a[i]));
  Button2.Visible := False;     // Приховуємо кнопку Button2
end;
```

Вище у опрацьовувачі події OnClick компонента Button1 для введення елементів масиву використовується функція InputBox. Якщо цю функцію не використовувати, то вказаний оброблювач події можна, наприклад, наповнити таким кодом (в цьому разі редактор використовується для введення всіх даних):

```
procedure TForm1.Button1Click(Sender: TObject);
var
  i: Integer;
begin
  if Tag = 0 then begin
    DecimalSeparator := '.';
    N := StrToInt(edInput1.Text);     // Кількість елементів
  end
  else
    a[Tag - 1] := StrToFloat(edInput1.Text);
  if Tag = N then begin
    mmOutPut1.Lines.Add('Початковий масив'); // Повідомлення
    for i := 0 to N - 1 do        // Виводимо початкові дані
      mmOutput1.Lines.Add('a[' + IntToStr(i + 1) + '] = ' +
                          FloatToStr(a[i]));
    lbOutput1.Visible := False;     // Мітка тепер прихована
    edInput1.Visible := False;    // Однорядковий редактор і
    Button1.Visible := False;      // Button1 тепер невидимі
    Button2.Visible := True;      // Button2 тепер є видимою
  end;
  lbOutput1.Caption := 'Уведіть елемент a['
                       + IntToStr(Tag + 1) + ']';
  Tag := Tag + 1;
end;
```

У наведеному тексті використана властивість Tag, що має тип Integer і є у кожного компонента (у наведеному вище коді ця властивість є властивостю форми). Вона має тип Integer, має за замовченням значення 0 і може використовуватися програмістом за власним розсудом. У наведеному вище програмному коді властивість Tag, з одного боку, використовується для вказівки того, що вводиться – розмір масиву (Tag = 0) або елементи масиву (Tag = 1). З другого боку, при введенні елементів масиву, ця властивість використовувалася для їх індексування. Якщо Tag отримує значення N, то здійснюється виведення початкового вмісту масиву.

Приклад 3. Дано натуральне число *n*, дійсне число *w* і масив з *n* дійсних чисел. Видалити з масиву елемент, що є найближчим до числа *w*.

Розмістимо на формі однорядковий редактор TEdit (надамо йому ім'я edInput1) для введення даних, багаторядковий редактор TMemo (надамо йому ім'я mmOutput1) для виведення результату та дві кнопки Button1 та Button2, перша з яких буде використовуватися для подання команд на введення даних, а друга – для подання команди на видалення вказаного в умові задачі елемента масиву і виведення його остаточного вмісту. Кнопки розмістимо одну на одній, причому у властивість Visible кнопки Button2 запишемо занчення False. Запишемо також у властивості Caption кнопок Button1 та Button2 тексти Ввести і Видалити відповідно. Над редактором edInput1 розмістимо мітку, надавши їй ім'я lbOutput1.

Включимо також в опис класу TForm1 у розділ **public** опис трьох полів:

```
    w: Real;                                    // Задане число
    N: Integer;                    //Кількість елементів масиву
    a: array [0..19] of Real;                                //Масив
procedure TForm1.FormCreate(Sender: TObject);
begin
  DecimalSeparator := '.';
  Tag := -1;
  lbOutput1.Caption := 'Уведіть число w';
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  i: Integer;
begin
  case Tag of
    -1: begin
          w := StrToFloat(edInput1.Text);       // Введення w
```

```
        lbOutput1.Caption := 'Уведіть кількість ' +
                             'елементів масиву';
      end;
    0 : begin
        N := StrToInt(edInput1.Text);        // Введення N
        lbOutput1.Caption := 'Уведіть елемент a['
                     + IntToStr(Tag + 1) + ']';
      end;
   else begin
        a[Tag] := StrToFloat(edInput1.Text);
        lbOutput1.Caption := 'Уведіть елемент a['
                     + IntToStr(Tag + 1) + ']';
        if Tag = N then begin              // Повідомлення
          mmOutPut1.Lines.Add('Початковий масив');
          for i := 1 to N do    // Виводимо початкові дані
           mmOutput1.Lines.Add('a[' + IntToStr(i) + '] = '
                              + FloatToStr(a[i]));
          lbOutput1.Visible := False;            // Мітка,
          edInput1.Visible := False;        // редактор та
          Button1.Visible := False;  // Button1 приховані,
          Button2.Visible := True;    // а Button2 видимая
        end;
      end;
  end;
  Tag := Tag + 1;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  i: Integer;
  k: Integer;           // Номер елемента, що видалятиметься
  Min: Real;                          // Мінімальна відстань
begin
  k := 1;
  Min := Abs(w - a[1]);
  for i := 2 to N do                  // Перебираємо елементи
    if Abs(w - a[i]) < Min then begin
      k := i;                // Запам'ятовуємо номер елемента
      Min := Abs(w - a[i]);          // та відстань до нього
    end;
  for i := k + 1 to N do
    a[i - 1] := a[i];                        // Зсув елументів
  Dec(N);                           // Розмір масиву зменшився
  mmOutPut1.Lines.Add('Результуючий масив');
  if N = 0 then                 // Результуючий масив порожній
    mmOutPut1.Lines.Add('порожній')
```

```
  else
    for i := 1 to N do            // Виведення вмісту массиву
      mmOutput1.Lines.Add('a[' + IntToStr(i) + '] = ' +
                          FloatToStr(a[i]));
  Button2.Visible := False;     // Приховуємо кнопку Button2
end;
```

3. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

За час, відведений для виконання лабораторної роботи (2 академічні години), студент повинен:

1. Розробити алгоритм розв'язання задачі, запропонованої для програмування.
2. Здійснити програмну реалізацію розробленого алгоритму.
3. Здійснити відлагодження програми, виправивши синтаксичні та логічні помилки.
4. Підібрати тестові дані для перевірки програми, включаючи виняткові випадки.
5. Оформити звіт до лабораторної роботи.
6. Відповісти на контрольні запитання.
7. Здати викладачу працездатну програму з демонстрацією її роботи на декількох варіантах вихідних даних.

4. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке масив?
2. Як описується одновимірний масив у Delphi?
3. Як задається діапазон змінення індексів у описі масивів?
4. Чи можна при задаванні діапазону зміни індексів використовувати вирази?
5. Чи можна задавати діапазон зміни індексів з використанням змінних?
6. Яке обмеження існує для типу меж діапазону?
7. Чи можна вмістити у масив дані різних типів?
8. Як здійснюється звертання до елементів масиву?
9. Чи можна при звертанні до єлементів масиву використовувати вирази?
10. Як можна описати масив у разі можливого набуття індексом всіх занчень деякого типу?
11. Яке обмеження існує для діапазону зміни індексу?
12. Яке обмеження існує для обсягу пам'яті, що може бути відведена під масив?

13. Змінні a та i мають такий опис:

```
var
  a: array [0..99] of Real;
  i: Real;
```

Маємо такий оператор:

```
for i := 0 to 99 do a[i] := i;
```

Чи є помилки у наведених вище фрагментах програмного коду? У разі позитивної відповіді зробіть пояснення.

14. Змінні a та i мають такий опис:

```
var
  a: array [100] of Real;
  i: Integer;
```

Маємо такий оператор:

```
for i := 1 to 100 do a[i] := i;
```

Чи є помилки у наведених вище фрагментах програмного коду? У разі позитивної відповіді зробіть пояснення.

15. Змінні a та i мають такий опис:

```
var
  a: array [0..99] of Real;
  i: Integer;
```

Маємо такий оператор:

```
for i := 1 to 100 do a[i] := i;
```

Чи є помилки у наведених вище фрагментах програмного коду? У разі позитивної відповіді зробіть пояснення.

16. Змінні a та i мають такий опис:

```
var
  a: array [0..99] of Boolean;
  i: Integer;
```

Маємо такий оператор:

```
a := True;
```

Чи є помилки у наведених вище фрагментах програмного коду? У разі позитивної відповіді зробіть пояснення.

17. Змінні a та i мають такий опис:

```
var
  a: array [-9..100] of Real;
  i: Integer;
```

Маємо такий оператор:

```
a := 1.4;
```

Чи є помилки у наведених вище фрагментах програмного коду? У разі позитивної відповіді зробіть пояснення.

18. Маємо такі описи:

```
var
  a: array [9..0] of Real;
  i: Integer;
```

У програмі зустрівся такий оператор:

```
for i := 9 downto 0 do a[i] := i;
```

Чи є помилки у наведених вище фрагментах програмного коду? У разі позитивної від-повіді зробіть пояснення.

19. Маємо такий опис:

```
var
  n, i: Integer;
```

У програмі зустрілася така послідовність операторів:

```
n := 20;
var
a: array [1..n] of Real;
for i := 1 to n do a[i] := i;
```

Чи є помилки у наведених вище фрагментах програмного коду? У разі позитивної відповіді зробіть пояснення.

20. Маємо такі описи:

```
var
  a, b: array [0..9] of Boolean;
  i, j: Integer;
```

Для переписування вмісту масива a у масив b використаний такий оператор:

```
for i := 0 to 9 do
  for j := i to i do
    b[i] := a[j];
```

Чи є помилки у наведених вище фрагментах програмного коду? Зробіть пояснення.

21. Маємо ті ж описи, що й у попередньому запитанні. Нижче наведений оператор подвійного циклу. Що можна саказати про нього?

```
for i := 0 to 9 do
  for j := 0 to i do
    b[i] := a[j];
```

5. ВАРІАНТИ ЗАДАЧ

1. Дано натуральне число n ($n \le 100$) і дійсні числа x_1, x_2, …, x_n. Знайти номер найменшого додатного числа в заданій послідовності. Якщо в послідовності x_1, x_2, …, x_n відсутні додатні числа, результатом повинне бути значення –1.
2. Дано натуральне число n ($n \le 100$) і послідовність дійсних чисел x_1, x_2, …, x_n. Якщо в результаті заміни від'ємних членів послідовності x_1, x_2, …, x_n їхніми квадратами члени нової послідовності будуть утворювати незростальну послідовність, то одержати добуток членів первинної послідовності; у супротивному випадку одержати їхню суму.
3. Дано натуральне числа n ($n \le 100$) і послідовність цілих чисел a_1, a_2, …, a_n. Поміняти місцями в цій послідовності найбільший і найменший члени. Якщо в послідовності кілька найбільших або найменших елементів, то розглядати останні з таких.
4. Дано натуральне число n ($n \le 100$) і масив з n дійсних чисел. Кожен елемент цього масиву замінити середнім арифметичним всіх передуючих йому елементів, включаючи його самого.
5. Дано натуральне число n ($n \le 100$) і послідовність з n дійсних чисел. Вивести всі члени послідовності, починаючи з максимального члена і закінчуючи мінімальним, поза залежністю від їх взаємного розташування (завжди першим повинен виводитись максимальний член, а останнім – мінімальний).
6. Дано масив з n цілих чисел, де n – задане натуральне число ($n \le 100$). Сформувати новий масив, що містить тільки ті елементи первинного масиву, які є простими числами.

7. Дано натуральне число n $(n \le 10)$ і цілі числа a_1, a_2, …, a_{3n}. З'ясувати, чи правда, що для всіх a_{n+1}, a_{n+2}, …, a_{3n} є рівні серед a_1, a_2, …, a_n.
8. Дано натуральне число n $(n \le 100)$ і послідовність цілих чисел a_1, a_2, …, a_n. Одержати нову послідовність, вилучивши з первинної послідовності всі члени зі значенням $\min(a_1, a_2, \ldots, a_n)$.
9. Дано натуральне число n $(n \le 100)$ і масив з n цілих чисел. Знайти найменше натуральне число, відсутнє в масиві.
10. Дано натуральне число n $(n \le 100)$ і послідовність відмінних від нуля цілих чисел a_1, a_2, …, a_n. Якщо в послідовності числа з різними знаками чергуються, вивести початкову послідовність. У супротивному випадку виключити з послідовності всі додатні члени послідовності.
11. Дано натуральне число n $(n \le 100)$ і цілі числа a_1, a_2, …, a_n. Якщо в послідовності a_1, a_2, …, a_n жодне парне число не розташоване після непарного, то одержати нову послідовність з усіх від'ємних членів вихідної послідовності. У супротивному випадку у нову послідовність вмістити всі додатні числа з послідовності a_1, a_2, …, a_n. В обох випадках порядок проходження чисел замінити на зворотний.
12. Дано натуральне число n $(n \le 100)$ і послідовність дійсних чисел a_1, a_2, …, a_n. Залишити без зміни послідовність a_1, a_2, …, a_n, якщо вона впорядкована за неубуванням або незростанням; у супротивному випадку видалити з неї ті члени, порядкові номери яких кратні п'яти, зберігши порядок залишених членів.
13. Дано натуральне число n $(n \le 100)$ і масив дійсних чисел a_1, a_2, …, a_n. Упорядкувати цей масив за незростанням. Скористатися таким методом. Знайти найменший елемент масиву і переставити його з першим елементом, потім серед елементів масиву, починаючи з другого, знайти найменший і переставити його з другим і т. д.
14. Дано натуральне число n $(n \le 100)$, ціле число m і масив з n цілих чисел. Чи міститься число m в масиві? Відповіддю повинне бути число 1 при позитивній відповіді і число 0 – при негативній.
15. Дано натуральне число n $(n \le 100)$ і дійсні числа a_1, a_2, …, a_n. Переставити члени послідовності a_1, a_2, …, a_n так, щоб спочатку розташувалися всі її невід'ємні члени, а потім всі від'ємні. При цьому повинен бути збережений початковий порядок, як серед всіх невід'ємних членів, так і серед від'ємних.

СПИСОК ЛІТЕРАТУРИ

1. Безменов, М. І. Турбо Паскаль 7.0 : навч. посіб. / М. І. Безменов. – Х. : НТУ «ХПІ»; Парус™, 2005. – 240 с.
2. Кэнту, М. Delphi 7 : Для профессионалов / М. Кэнту – СПб. : Питер, 2004. – 1101 с.
3. Архангельский, А. Я. Программирование в Delphi 6 / А. Я. Архангельский. – М. : БИНОМ, 2002. – 1120 с.
4. Дарахвелидзе, П. Г. Программирование в Delphi 7 / П. Г. Дарахвелидзе, Е. П. Марков. – СПб. : БХВ-Петербург, 2003. – 784 с.
5. Культин, Н. Б. Основы программирования в Delphi 7 / Н. Б. Культин. – СПб.: БХВ-Петербург, 2003. – 608 с.
6. Пестриков, В. М. Delphi на примерах / В. М. Пестриков, А. Н. Маслобоев. – СПб. : БХВ-Петербург, 2005. – 496 с.
7. Ремкеев, А. А. Курс Delphi для начинающих. Полигон нестандартных задач / А. А. Ремкеев, С. В. Федотова. – М. : СОЛОН-Пресс, 2006. – 360 с.
8. Митчелл, К. Керман. Программирование и отладка в Delphi™ : учебный курс / Митчелл К. Керман. – М. : Вильямс, 2004. – 720 с.
9. Парижский, С. М. Delphi : Только практика / С. М. Парижский. – К. : МК-Пресс, 2005. – 208 с.
10. Культин, Н. Б. Основы программирования в Delphi 2007 / Н. Б. Культин. – СПб. : БХВ-Петербург, 2008. – 480 с.

Навчальне видання

Методичні вказівки
до лабораторної роботи
«Використання одновимірних масивів у програмах мовою Delphi»
з курсу «Програмування» для студентів напряму 6.040302 – Інформатика
(спеціалізація «Соціальна інформатика»)

Укладач БЕЗМЕНОВ Микола Іванович

Відповідальний за випуск О. С. Куценко
Роботу до видання рекомендував О. В. Горелий

За авторською редакцією

План 2010 р., поз. 15 / 45-10

Підп. до друку 15.03.2010 р. Формат 60 × 84 $^{1}/_{16}$. Папір офісний.
Riso-друк. Гарнітура Таймс. Ум. друк. арк. 0,9. Наклад 50 прим.
Зам. № 67. Ціна договірна.

Видавничий центр НТУ «ХПІ».
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21

Друкарня НТУ «ХПІ», 61002, Харків, вул. Фрунзе, 21